Population of blocks

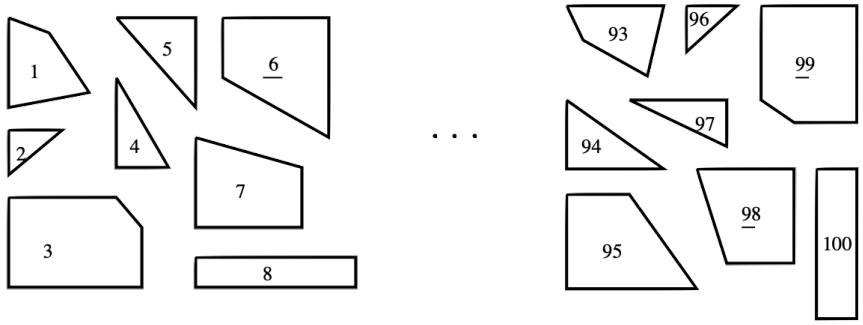
```
## Set this up for your own directory
imageDirectory <- "MyAssignmentDirectory/img"  # e.g. in current "./img"
dataDirectory <- "MyAssignmentDirectory/data"  # e.g. in current "./data"
path_concat <- function(path1, path2, sep="/") paste(path1, path2, sep = sep)
```

---

50 marks

Consider a study population $\mathcal{P}_{Study}$ consisting of $N = 100$ blocks labelled $u = 1, 2, 3, \ldots, 100$.

The blocks are of uniform thickness and density (all blocks were cut from the same opaque plastic sheet of about 5 mm thickness), but have different shapes such as those shown below:



Data on this population of 100 blocks are available as an R data set `blocks`. This data set has four variates: block `id` number, `weight` in grams, `perimeter` in centimetres, and the `group` of the block (being either `A` or `B`). It can be loaded from the assignment data directory as follows:

```
load(path_concat(dataDirectory, "blocks.rda"))
head(blocks, n = 3)
```

```
##   id weight perimeter group
## 1  1     55        32     B
## 2  2     35        27     B
## 3  3     35        25     A
```

In this question, you will examine different possible attributes of interest for this population.

a. Simple numerical attributes.

   i. *(1 mark)* Summarize this population by the following attributes on the variates `weight` and `perimeter`: the population median, mean, and standard deviation (here computed using `sd()` with denominator $N-1$).

   ii. *(1 mark)* Repeat the above summaries but now conditional on the group to which each block belongs. Now include the number in each group.

   iii. *(3 marks)* On the basis of the above computed attributes, describe how each group differs from the whole population and from each other.

b. Simple graphical attributes.

   i. *(8 marks)* Draw (suitably labelled) histograms of the weight for the whole population, only the blocks in group `A`, and only the blocks in group `B`. Make sure you use the same `xlim = extendrange(blocks$weight)`, the same `ylim = c(0,20)`, and the same `breaks <- seq(min(xlim), max(xlim), length.out = 20)` in all histograms. Add a vertical dashed red line at the average of the blocks in each case. Arrange the three plots so that they appear above one another in your display.

   Comment on the differences between histograms.

   Show your code.

   ii. *(6 marks)* Using formula notation, draw pairs of (suitably labelled) boxplots comparing the two groups, first with respect to a difference in block weights and then with respect to the perimeters. Comment on how the two groups compare.

   Show your code.

   iii. *(9 marks)* Quantile plots. A sample quantile plot is a scatterplot of the point pairs

$$(\frac{i-0.5}{n}, y_{(i)})$$

   for $i = 1, 2, \ldots, n$ and $y_{(1)} \leq y_{(2)} \leq \cdots \leq y_{(n)}$ is the sample *order statistic*. The values $\frac{i-0.5}{n}$ are returned by the R function `ppoints()`.

   On a single plot, overlay the sample quantiles for the perimeters of all the blocks, of the perimeters of those in group `A`, and of those in group `B`. Use different colours and point symbols for each set. Add a legend to distinguish the groups.

   The three groups of points trace out different curves.

   - what does a difference in heights of the curves near the middle show?
   - what does a difference in the slopes of the curves near the middle tell you?

   Comment on the differences between the three groups.

   Show your code.

c. *(5 marks)* Scatterplots. In loon , use `l_plot()` to plot `weight` versus `perimeter` (i.e. the pairs `(perimeter, weight)`); make sure you save the plot on an R variable (e.g. `p <- l_plot(...)`).

   Then,

   - select all of the points in the plot and jitter them
     - this can be done programmatically using `p["selected"] <- TRUE`
     - and `l_move_jitter(p)`
   - deselect the points programmatically
   - print the plot as part of your assignment
   - swap the axes programmatically and again print the plot
   - note: `grid.arrange()` from the gridExtra package can be used to place the two plots side by side in your output

Show your code.

Describe how these two variables appear to be related (if at all).

d. Attributes given by fitted models.

   i. *(5 marks)* Find the simplest polynomial that fits the `weight` as a function of `perimeter` for these blocks.

      Show your code for fitting (only) the final model you choose and print a summary of the fitted model.

      Using the `fit` of your model examine (and submit) the two plots from `plot(fit, which = c(1,2))`.

      Describe the model you have selected and common on the quality of its fit.

   ii. *(5 marks)* Use the `power_xy()` function (from the course slides) on the perimeter (as x) and weight (as y). Find values of α_x and α_y on Tukey's ladder that make the least-squares line a plausible summary.

      Use these values with `lm()` to fit the model appearing as the straight line in your transformed plot. Note, if you have an integer power for `perimeter`, fit the model using `poly()` with degree equal to the integer power.

      Show a summary of your fitted model and comment on the quality of its fit.

   iii. *(3 marks)* Which of the two previous fitted models do you prefer? Explain why you choose that model.

   iv. *(4 marks)* Explain why a model relating weight as a function only of perimeter makes no physical sense. How might the concept of study error be used to describe the problem?

      If, for example, our target population were to include all shaped blocks then any model summary based on the relationship between weight and perimeter would be quite different for our study population of convex blocks. This constitutes study error.